



ÉCOLE NATIONALE DES SCIENCES GÉOGRAPHIQUES

Rapport d'analyse
Projet d'application JPO
Explor'Descartes

Auteurs :

Lucas Tacito
Émilie Jochel
Hugo De Paulis
Wandrille Falala

Commanditaire :
Mandine Javerzac

Encadré par :
Miryam Morand
Maelys Gaillet

Table des matières

1	Contexte du projet Explor'Descartes	1
1.1	Présentation du projet	1
1.2	Enjeux du projet	1
1.3	Évaluation financière du projet	1
1.4	Aspects sociaux du projets	2
2	Objectifs de l'étude	3
2.1	Objectifs de l'analyse	3
2.2	Acteurs du projet	3
2.3	Besoins des commanditaires	3
2.3.1	Partie visiteur	3
2.3.2	Partie administrateur	4
2.4	Contraintes	4
2.5	Diagrammes de cas d'utilisations	4
3	Analyse fonctionnelle	6
3.1	Diagramme d'activités	6
3.2	Diagramme de classes	7
3.3	Relation Base de Données - Application	8
4	Étude technique du projet	9
4.1	Choix des logiciels	9
4.1.1	Choix des langages de programmations	9
4.1.2	Choix de l'API cartographique	9
4.1.3	Choix du Système de Gestion de Base de Donnée (SGBD)	9
4.2	Architecture du projet	10
5	Suivi de projet	11
5.1	Choix du nom et du logo du projet	11
5.1.1	Nom du projet	11
5.1.2	Logo du projet	11
5.2	Les risques du projet	12
5.3	Liste des livrables	12
5.4	Planning prévisionnel	12
5.5	Liens utiles	14

Chapitre 1

Contexte du projet Explor'Descartes

1.1 Présentation du projet

L'université Paris-Est Marne-la-Vallée changera de nom le 1er janvier 2020 et fusionnera avec l'IFSTTAR et 4 grandes écoles : l'ESIEE, l'EIVP, l'ENSAVT et l'ENSG pour devenir une université à vocation internationale : l'Université Gustave-Eiffel.

Dans ce contexte de fusion, le projet a pour but de fournir une application pour les visiteurs venant assister aux Journées Portes Ouvertes (JPO) du campus de la Cité Descartes afin de les aider à profiter au maximum de ces journées. Cette application devra faciliter la visite de l'utilisateur en proposant différentes fonctionnalités d'organisation, de géolocalisation et d'affichage d'informations.

Les commanditaires de ce projet sont Mandine Jarverzac, responsable de l'événementiel et de la communication rédactionnelle de l'Université Paris-Est Marne-la-Vallée et coordinatrice JPO de la Cité Descartes et l'ensembles des responsables communication de chaque établissements de la Cité Descartes.

1.2 Enjeux du projet

Ce projet va devoir répondre à plusieurs enjeux :

- faciliter l'organisation des journées portes ouvertes ;
- atteindre un public plus grand pour les portes ouvertes ;
- affirmer le rôle de l'ENSG en tant qu'école d'informatique et géomatique au sein du campus Descartes.

1.3 Évaluation financière du projet

En se basant sur le nombre de séances réservées pour la réalisation du projet ainsi qu'au choix des logiciels (cf-), il est possibles de réaliser une évaluation financière du projet.

	Nombre d'heures	Nombre de personnes	Total (heures)
Phase d'analyse	4 x 6h	4	96h
Développement	12 x 6h	4	288h
Total d'heures travaillées			384h

Si les commanditaires avaient eu recours à un prestataire extérieurs se basant sur le même nombre d'heures de réalisation du projet, ils auraient dû payer environ 400 heures travaillées en comptant les réunions avec les commanditaires.

Coûts matériels :

Serveur	1€ à 5€ par mois
API MapBox	gratuit tant que le nombre d'utilisateurs et de requêtes n'excède pas 50000 par mois, sinon il faut payer 0.45\$ pour 500 utilisateurs et/ou 1000 requêtes supplémentaires par mois.

Les principaux coûts du projet sont donc liés à la partie technique du projet.

1.4 Aspects sociaux du projets

Pour mener à bien ce projet, nous avons établis un lien très régulier avec les commanditaires grâce à de nombreux rendez vous prévus pour suivre l'avancement du projet.

De plus, Myriam Morand, responsable communication de l'ENSG, ainsi que Maelys Gaillet, étudiante en Master Communication encadrent notre équipe pour mener à bien le projet.

Nous avons désigné Émilie Jochel comme cheffe de projet afin de coordonner nos actions, de résoudre les potentiels conflits et de trancher dans le processus de décision.

Chapitre 2

Objectifs de l'étude

2.1 Objectifs de l'analyse

Notre travail lors de la phase d'analyse a eu pour but de dégager plusieurs objectifs :

- Définir les besoins des acteurs du projet et constituer un cahier des charges ;
- Définir les fonctionnalités de l'application, l'interface utilisateur ;
- Définir les fonctionnalités de l'interface administrateur ;
- Définir l'architecture technique de l'application ;

2.2 Acteurs du projet

Les acteurs du projets sont les suivants :

- les **commanditaires** : Mandine Jarverzac représentant l'UPEM et les autres membres du groupe d'établissements participant à la JPO de la cité Descartes ;
- le **personnel encadrant de l'ENSG** : Myriam Morand, responsable communication de l'ENSG, et Maelys Gaillet, étudiante en Master 2 communication Web ainsi que les professeurs référents du DEI ;
- les **administrateurs** des bases de données qui se chargeront de les remplir avec les bonnes informations sur les établissements et de récupérer les formulaires de satisfactions ;
- les **visiteurs** des Journées Portes Ouvertes.

2.3 Besoins des commanditaires

Les besoins des commanditaires s'articulent sur deux aspects :

- la partie **visiteur** de la JPO, c'est à dire l'application Android ;
- la partie **administrateur** de base de données, c'est à dire une interface Web.

2.3.1 Partie visiteur

L'application, partie visiteur du projet, devra répondre à des besoins étalés sur trois périodes de temps :

- la préparation **avant** les JPO ;
- une utilisation **pendant** les JPO ;
- un retour utilisateur **après** les JPO.

L'utilisateur pourra utiliser cette application en **mode connecté** afin de répondre à des formulaire et ajouter des favoris ou bien en **mode hors-connexion**.

Cette application vise à aider l'utilisateur à organiser sa journée **en avance** en fonction des événements et conférences qui seront disponibles dans les différents établissements du campus. L'utilisateur pourra ainsi organiser son **planning**, sélectionner ses **événements favoris** et être alerté à l'approche d'un événement grâce à un **système de notification**.

L'application devra également guider et assister le visiteur pendant sa visite sur le campus grâce à une **carte** comportant une fonction d'**itinéraire**, les moyens de **transport disponibles**, les **restaurants** ainsi que des **marqueurs pour chaque établissement**. Ces marqueurs redirigeront l'utilisateur vers les **formations** des différents établissements, les **stands** présents à cet endroit, les **conférences** qui y sont organisées ainsi que le **site internet de l'établissement** pour des informations plus exhaustives.

Un dernier aspect de l'application est de permettre à l'utilisateur de soumettre un formulaire de satisfaction afin d'évaluer les événements **après** y avoir assisté.

2.3.2 Partie administrateur

Les commanditaires du projets nous ont aussi demandé de concevoir une interface pour les administrateurs des bases de données. Ces administrateurs n'ayant pas de connaissances poussés avec les outils de SGBD tels que Postgres, PgAdmin... il est nécessaire de réaliser une **interface** de gestion de base de données **simple d'utilisation** et dont les fonctionnalités restent **basiques**.

Ces administrateurs doivent pouvoir **ajouter**, **modifier** et **supprimer** des lignes dans les base de données.

2.4 Contraintes

Le projet présente plusieurs contraintes :

- L'application doit pouvoir **remplir les fonctions et objectifs définis** dans le cahier des charges réalisé en concertation avec les commanditaires du projet ;
- L'utilisation des données des utilisateurs de l'application et des sondages préalablement effectués doivent être **conforme à la réglementation définie par la CNIL et le Règlement Général sur la Protection des Données (RGPD)** ;
- La charte graphique doit être **esthétique et cohérente** pour convenir aux visiteurs et donc aux commanditaires ;
- Un **accès** à la **base de données** et au **code** de l'application doit être garanti afin de permettre des évolutions postérieures au projet. Une documentation utilisateur et administrateur détaillée est donc nécessaire. Une interface pour les administrateurs est aussi nécessaire ;
- Il est aussi nécessaire de s'assurer de la **disponibilité des serveurs distants** afin de pouvoir héberger les bases de données.

2.5 Diagrammes de cas d'utilisations

La description des diagrammes correspond aux parties relatives aux besoins des commanditaires.

Le visiteur, lorsqu'il ouvre l'application pour la première fois, va devoir se connecter avec un compte déjà existant ou bien s'enregistrer s'il veut pouvoir profiter des toutes les fonctionnalités de l'application. Il pourra s'enregistrer en remplissant un formulaire ou en utilisant un réseau social (Facebook, Instagram ou Google). Son adresse mail et son mot de passe seront donc enregistrer dans la base de données Visiteurs. Le visiteur peut aussi utiliser l'application en mode hors-connexion mais il n'aura pas accès à toutes les fonctionnalités telles que l'ajout de favoris ou le remplissage du formulaire de satisfaction.

Il pourra ensuite accéder aux différentes fonctionnalités telles que la navigation sur la carte, l'affichage du planning des JPO, l'affichage des établissements, etc... Pour réaliser cela, le visiteurs interagira avec la base de données.

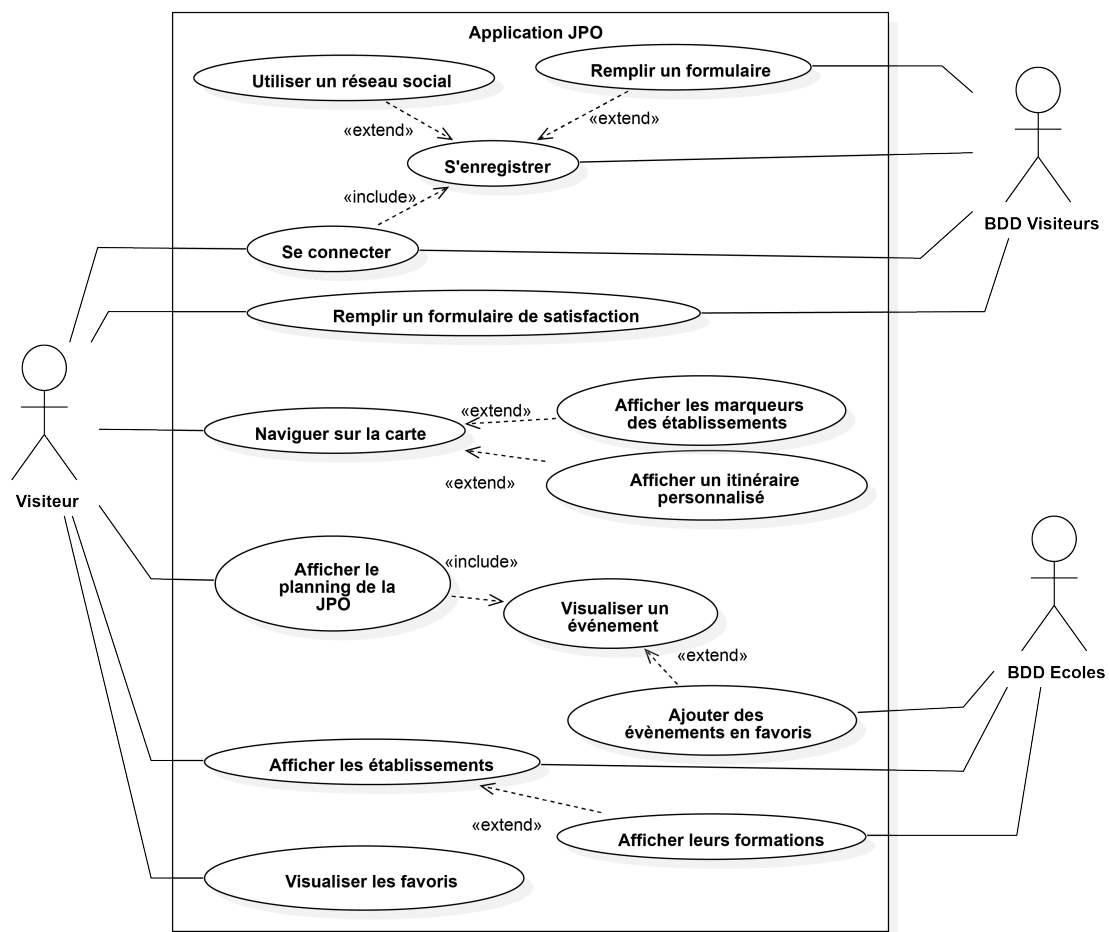


FIGURE 2.1 – Diagramme de cas d'utilisation visiteur

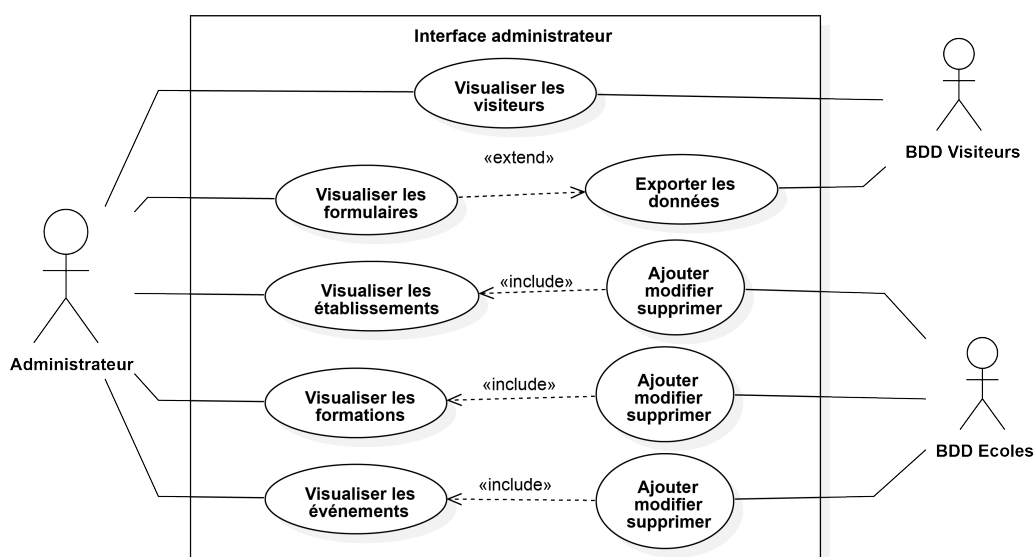


FIGURE 2.2 – Diagramme de cas d'utilisations administrateur

Chapitre 3

Analyse fonctionnelle

Dans cette partie nous allons décrire les grandes fonctionnalités de l'application destinée au visiteur. Nous n'allons pas décrire l'interface administrateur car il s'agit d'une interface web basique servant seulement à simplifier la réalisation de requêtes SQL pour les administrateurs.

3.1 Diagramme d'activités

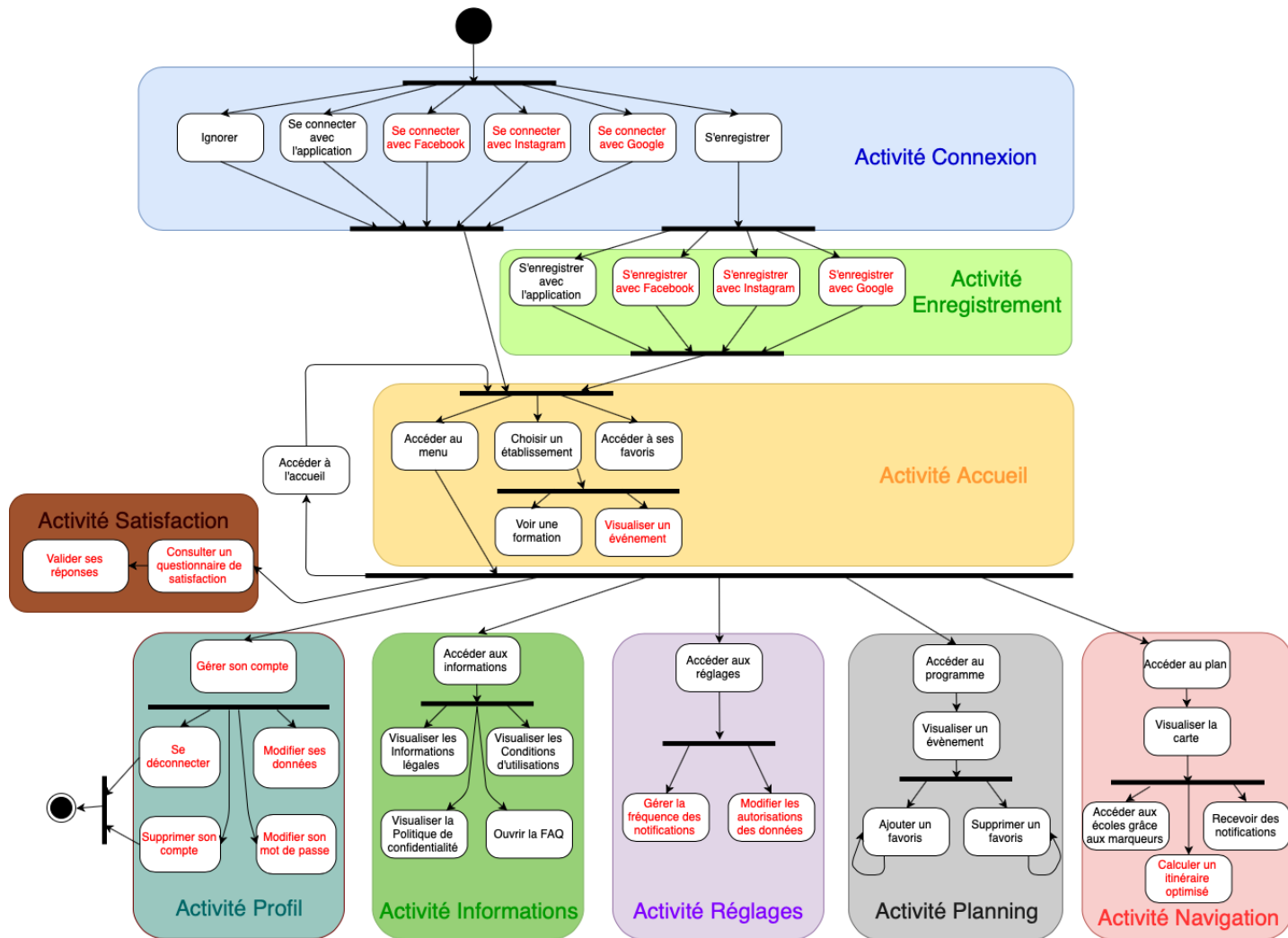


FIGURE 3.1 – Diagramme d'activité de l'application

Le diagramme d'activité décrit l'utilisation de l'application par un visiteur en suivant les besoins définis par les commanditaires (cf-2.3.1). À cela, on ajoute les activités de visionnage d'informations légales et réglages de l'application. Chaque couleur indique une nouvelle page ouverte dans l'application.

3.2 Diagramme de classes

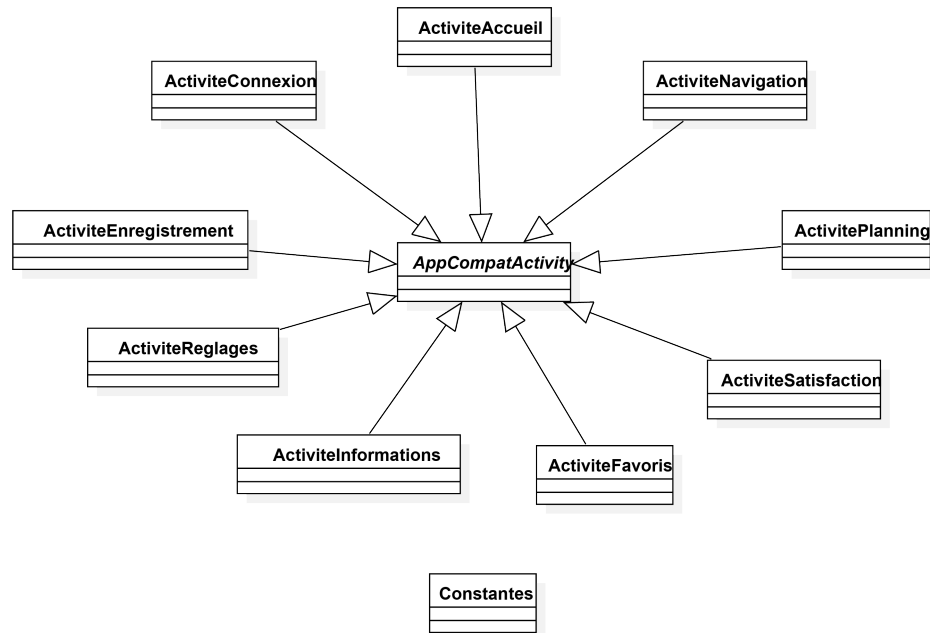


FIGURE 3.2 – Diagramme de classes

Pour coder notre application, nous allons donc créer onze classes, comme indiqué sur le diagramme de classe ci-dessus. Chaque classe correspond à une page en Android, à l'exception des classes Constante et de AppCompatActivity.

La classe principale **AppCompatActivity** rassemblera les diverses fonctionnalités de notre application et assurera le fonctionnement de l'application et de ses différentes parties.

ActiviteConnexion est la page que l'on verra en ouvrant l'application pour la première fois, avec un bouton de connexion pour se connecter par voie classique (pseudo et mot de passe) ou via les réseaux sociaux (Instagram, Facebook ou Google), pour créer un compte si on n'en possède pas déjà un, ou pour ignorer la phase de connexion, ce qui désactivera les fonctionnalités de favoris. **ActiviteEnregistrement** est la page qui permet de s'enregistrer et de créer un compte directement sur l'application, avec un pseudo et un mot de passe. **ActiviteAccueil** est le menu principal que l'on atteindra en se connectant sur l'application, avec un accès aux autres pages de l'application.

L'application en elle-même comprendra ensuite six pages principales, correspondant chacune à une fonctionnalité précise. **ActiviteNavigation** est la classe contenant la carte du campus permettant un repérage des environs, le calcul d'un itinéraire et l'affichage des marqueurs des établissements avec de plus amples informations sur l'école si l'on décide de cliquer sur le marqueur. **ActivitePlanning** contient le planning que l'on peut organiser pour sa JPO, avec la liste des événements prévus pour le remplir. Cette page enverra également des notifications à l'approche d'un événement enregistré dans le planning. **ActiviteFavoris** sera un menu dépliant présent sur le côté de l'écran de l'application et consultable à tout moment, contenant les établissements sélectionnés ayant retenu l'attention de l'utilisateur, dans lequel on pourra ajouter de nouveaux établissements.

ActiviteInformations est la page recensant les diverses informations sur les établissements et les formations proposées, que l'utilisateur pourra consulter pour définir ses priorités en fonction de ses centres d'intérêt. Les informations seront celles entrées dans la BDD établissement, elles pourront donc être mises à jour par les administrateurs. **ActiviteReglages** permet de changer divers paramètres dans l'application, comme la fréquence des notifications ainsi que les autorisations d'accès de l'application. Pour ce qui concerne l'enquête de satisfaction à remplir en fin de JPO, nous avons créé la classe **ActiviteSatisfaction**. Comme précisé précédemment, les informations rentrés par les visiteurs dans les pages de connexion et dans le formulaire de satisfaction seront accessibles aux administrateurs de l'application via la **BDD Visiteurs**.

Enfin, la classe **Constantes** contiendra diverses constantes dont nous pourrions avoir besoin pour le bon fonctionnement de l'application, sans passer par une base de données pour les stocker.

3.3 Relation Base de Données - Application

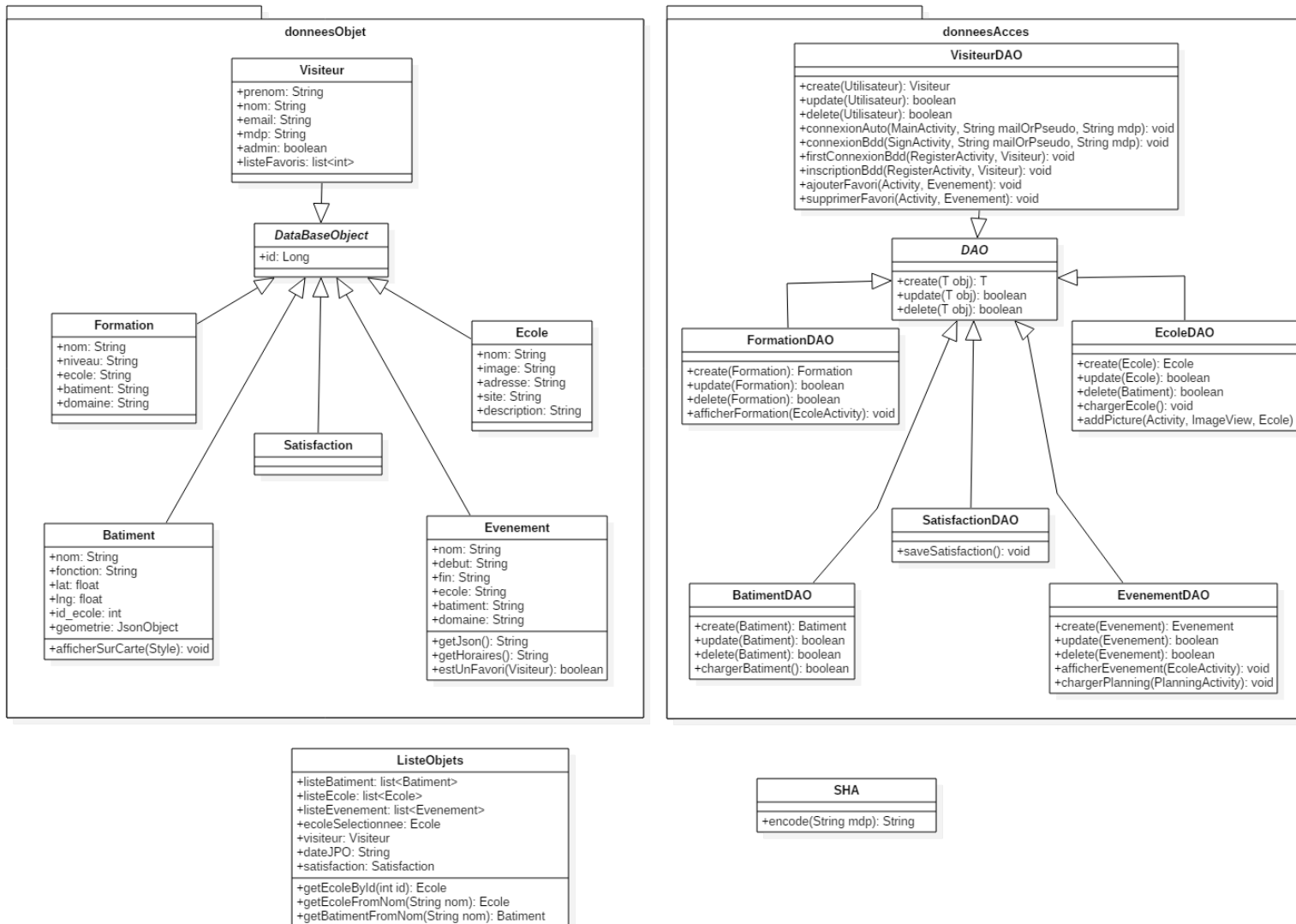


FIGURE 3.3 – Diagramme de classes DAO

Notre application devra pouvoir se connecter à différentes bases de données afin d'ajouter, modifier ou supprimer un objet. Elle doit aussi pouvoir retrouver des objets à partir d'identifiants pour implémenter leur version JAVA. Ne possédant de serveur définitif en début de projet, il faut donc une architecture permettant facilement la modification des accesseurs. Nous avons ainsi choisi le pattern DAO (ou Data Access Object) qui dissocie les classes des objets métiers et les classes des accesseurs. De cette manière, les objets métiers ne sont pas dépendants de la façon dont sont stockées les données.

Ainsi, dans notre architecture, la gestion de la base de données est séparée en deux packages :

- le **package donneesObjet** qui contient toutes les classes d'objets métiers, héritant d'une même classe abstraite **DataBaseObject** ;
- le **package donneesAcces** qui contient toutes les classes des accesseurs, héritant d'une même classe abstraite **DAO**.

Chapitre 4

Étude technique du projet

4.1 Choix des logiciels

4.1.1 Choix des langages de programmations

Les commanditaires ont exigé une application mobile comme rendu pour les visiteurs des Portes Ouvertes ainsi qu’une interface de gestion de base de donnée pour les administrateurs.

Nous avons donc fait le choix de coder en **Java** à l’aide du logiciel **Android Studio** pour la partie application car nous avons déjà utilisé cet outils lors de cours sur les applications mobiles Android.

Pour l’interface administrateur, nous avons décidé de créer une page Web en **HTML/CSS**, **Javascript** et **PHP** pour interagir avec les bases de données.

4.1.2 Choix de l’API cartographique

Le choix de l’API cartographique est très important car l’aspect géographique est au cœur de l’application. Elle doit respecter plusieurs impératifs tels que la gratuité et l’accès à une documentation complète. De plus, l’application doit pouvoir être reprise par une autre équipe de développement, elle doit donc être simple et connue dans le milieu du développement. Après avoir recherché différents SDK, nous avons finalement retenu Google Maps, MapBox et MapQuest. Aucun des trois n’est complètement libre, mais ils présentent un certain volume de données gratuits, suffisant dans le cadre de notre projet.

	Nombre de services gratuits	Documentation	Communauté	Qualité du service	Facilité d’utilisation
Google Maps	*	***	***	***	***
MapBox	***	***	**	***	**
MapQuest	**	**	*	**	***

TABLE 4.1 – Comparaison des API cartographiques

Pour être certain de ne pas dépasser la limite des fonctionnalités gratuites autorisée par l’API, notre choix s’est porté sur MapBox, car elle propose un gros volume de données (50000 utilisateurs et requêtes), correspondant ainsi à nos attentes. De plus, la qualité du service est équivalente à celle de Google Maps et la documentation est de très bonne qualité.

4.1.3 Choix du Système de Gestion de Base de Donnée (SGBD)

Pour le choix du SGBD, nous nous sommes intéressés au volume de données que les différents SGBD pouvaient traiter. La gratuité était également un impératif. Nous nous sommes donc retrouvé à comparer les deux services libres les plus connus : MySQL et PostgreSQL.

	Volume de données traités	Documentation	Communauté	Fonctionnalités	Respect de la norme SQL2003
PostgreSQL	***	***	***	***	***
MySQL	**	**	**	**	*

TABLE 4.2 – Comparaison des SGBD

Devant l'analyse de ces différents paramètres, nous avons donc sélectionné PostgreSQL qui de plus nous est familier.

4.2 Architecture du projet

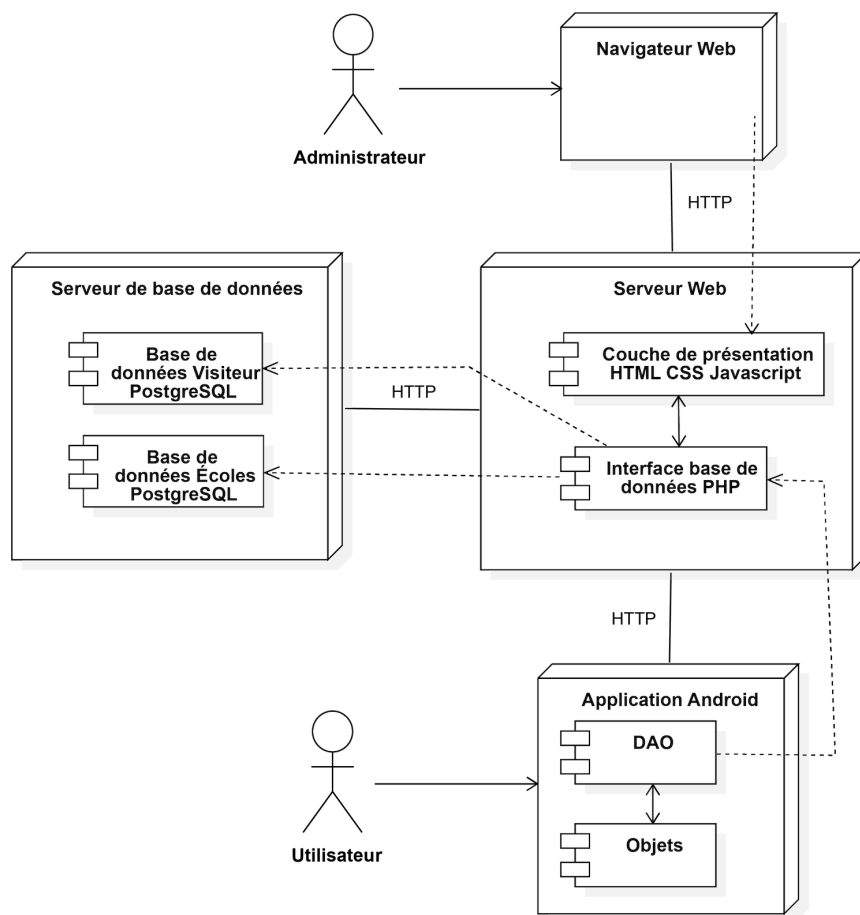


FIGURE 4.1 – Diagramme de déploiement

Le diagramme de déploiement ci-dessus décrit l'architecture de la solution proposée pour le projet. Le diagramme représente les interactions entre les différents appareils et leurs composants ainsi que les langages utilisés.

Il sera donc nécessaire d'utiliser deux serveurs de déploiements :

- Le serveur de base de données ;
- Le serveur Web qui hébergera l'interface administrateur.

L'application en elle-même pourra être diffusée grâce à Google Play.

Chapitre 5

Suivi de projet

5.1 Choix du nom et du logo du projet

5.1.1 Nom du projet

Les commanditaires souhaitaient que le nom de l'application contienne à la fois le nom du campus et la dimension de géographique de l'application. Après concertation avec les commanditaires, nous avons établi une liste de nom possibles pour l'application et le projet. Nous avons réalisé un sondage auprès des élèves de l'ENSG afin de sélectionner les noms les plus pertinents.

Finalement parmi les noms restants nous avons choisi «**Explor'Descartes**». Ce nom évoque la découverte et l'exploration du campus Descartes lors des Journées Portes Ouvertes.

5.1.2 Logo du projet

Comme pour le nom, les commanditaires avaient déjà préparé des pistes de réflexion pour le logo. Ils désiraient un logo qui évoque l'idée de géolocalisation et la Cité Descartes.

Pour évoquer l'identité de la Cité Descartes, les commanditaires nous ont conseillés d'utiliser la forme de l'oeuvre d'art du rond point devant l'ESIEE. Cette oeuvre représente l'axe de rotation de la Terre grâce à un grand pic de métal. Nous avons décidé de représenter l'aspect de géolocalisation par un marqueur classique utilisé dans les services de géolocalisation de type Google Maps ou Leaflet.

Enfin, le logo que nous avons retenu est tricolore, nous avons utilisé la couleur du nouveau logo de l'Université Gustave-Eiffel comme couleur principale.



FIGURE 5.1 – Logo du projet Explor'Descartes

5.2 Les risques du projet

Les risques qui pourraient compromettre l'aboutissement du projet sont décrits dans la matrice de risque ci-dessous :

Nature de risque	Description	Gravité 1-4	Probabilité	Actions préventives
Juridique	Risques légaux (CNIL)	4	Très faible	Respecter les préconisations de la CNIL
Intrinsèque à la gestion de projet	Application ne répondant pas aux attentes des utilisateurs	3	Faible	Réunions régulières avec les commanditaires, formulaires, définitions des besoins
Humain	Mésententes au sein de l'équipe	1	Faible	Communication au sein de l'équipe, chef de projet
Intrinsèque à la gestion de projet	Donnée non conformes pour les écoles	2	Faible	Modifications de la base de données, accès administrateur pour les responsables comm des écoles
Technique	Crash serveur	2	Moyenne	Méthodes de restauration de serveurs vues en cours de méthode agile, archive
Financier	Modifications des tarifs des logiciels utilisés	3	Faible	Logiciels de remplacement à utiliser
Délais	Non respect des délais	3	Moyenne	Priorisation des fonctionnalités non essentielles du planning, décrit dans le GANTT

FIGURE 5.2 – Matrice des risques du projet

5.3 Liste des livrables

- Application mobile
- Base de données Visiteurs
- Base de données Ecoles
- Interface web pour la mise à jour des BDD Ecole et Visiteurs pour les administrateurs
- Documentation relative à l'utilisation des bases de données
- Documentation pour l'installation de l'application
- Documentation visiteur (FAQ, video de présentation)
- READ ME
- Rapport d'analyse
- Compte-rendu du COPIL

5.4 Planning prévisionnel

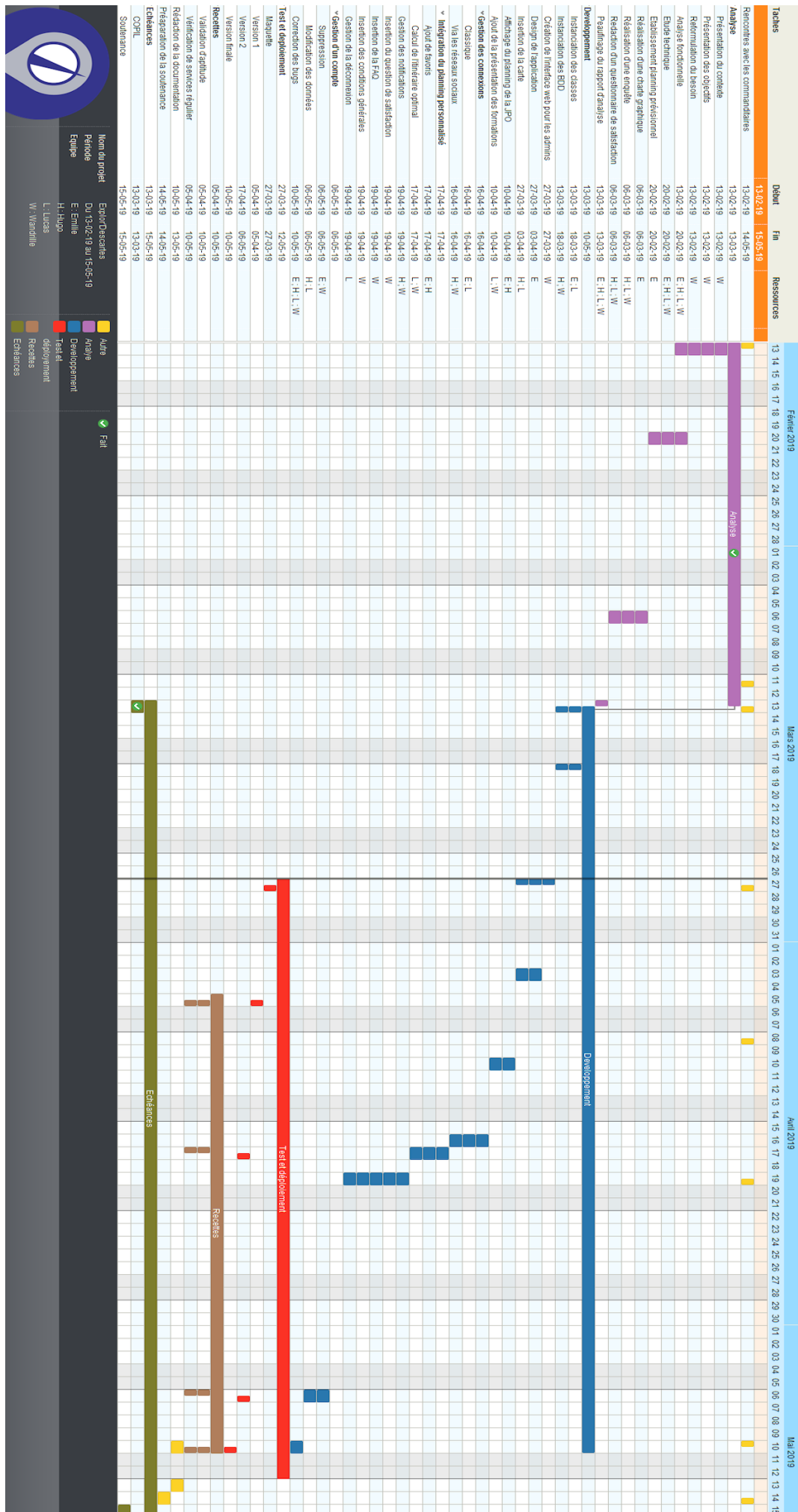
Nous avons réalisé un diagramme de GANTT afin de gérer au mieux l'organisation de notre projet.

Dans un premier temps, nous avons programmé les tâches relatives à la phase d'analyse. On y retrouve également un planning prévisionnel des étapes de développement. Pour cette partie nous avons choisi de programmer les fonctionnalités par ordre de priorité tout en prenant en compte la dépendance des tâches entre-elles.

En ce qui concerne, la répartition des travaux, nous avons tenté de maximiser le travail en paires. Cela permettra donc une meilleure cohésion de l'équipe tout en favorisant la production de meilleurs codes.

D'autre part, nous avons opté pour un déploiement de plusieurs versions tests afin de partager l'avancée de nos travaux avec nos commanditaires et ainsi collecter le maximum de retour de leur part, mais aussi sur un échantillon d'élèves.

Enfin, nous avons planifié quelques rencontres avec nos commanditaires afin de maintenir des échanges et bilans réguliers. Nous avons aussi programmé des étapes de recette, de sorte à assurer la conformité de l'application aux spécifications.



5.5 Liens utiles

Github du projet : https://github.com/Bamzuki/Projet_JP0

Site internet de l'Univeristé Paris-Est Marne-la-Vallée : <http://www.u-pem.fr>

Site internet des JPO de la Cité Descartes : <http://jpo.citedescartes.fr>